IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 11, 2019, accepted December 19, 2019, date of publication December 23, 2019,
date of current version December 31, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2961559*

# A Hybrid Markov and LSTM Model for Indoor Location Prediction

**PEIXIAO WANG**[1,3], **HONGEN WANG**[4], **HENGCAI ZHANG**[2,3], **FENG LU**[2,3], **AND SHENG WU**[1,3]

[1]The Academy of Digital China, Fuzhou University, Fuzhou 350002, China
[2]State Key Laboratory of Resources and Environmental Information System, IGSNRR, CAS, Beijing 100101, China
[3]Fujian Collaborative Innovation Center for Big Data Applications in Governments, Fuzhou University, Fuzhou 350002, China
[4]College of Geomatics, Shandong University of Science and Technology, Qingdao 266590, China

Corresponding authors: Hengcai Zhang (zhanghc@lreis.ac.cn) and Sheng Wu (wusheng@fzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 41771436 and Grant 41701521, in part by the National Key Research and Development Program of China, under Grant 2016YFB0502104 and Grant 2017YFB0503500, and in part by the Digital Fujian Program under Grant 2016-23.

**ABSTRACT** Accurate and robust indoor location prediction plays an important role in indoor location services. Markov chains (MCs) have been widely adopted for location prediction due to their strong interpretability. However, multi-order Markov chains ($k$-MCs) are not suitable for predicting long sequences due to problems of dimensionality. This study proposes a hybrid Markov model for location prediction that integrates a long short-term memory model (LSTM); this hybrid model is referred to as the Markov-LSTM. First, a multi-step Markov transition matrix is defined to decompose the $k$-MC into multiple first-order MCs. The LSTM is then introduced to combine multiple first-order MCs to improve prediction performance. Extensive experiments are conducted using real indoor Wi-Fi positioning datasets collected in a shopping mall. The results show that the Markov-LSTM model significantly outperforms five existing baseline methods in terms of its predictive performance.

**INDEX TERMS** Indoor location prediction, movement trajectory, Markov-LSTM.

## I. INTRODUCTION

In recent years, traditional "brick-and-mortar" industries have been severely affected by rapid developments in e-commerce [1]. Therefore, the traditional "brick-and-mortar" industries urgently need to find ways to help merchants establish relationships with customers and provide them with personalized shopping experience to improve the marketing ability of brick-and-mortar industries [2]. With the development of indoor positioning technology and the popularization of mobile terminal devices, there has been an explosive growth in the availability of indoor mobile trajectory data [3]. Such data are an important basis for indoor location services and provide new opportunities for the development of these industries [4], [5].

Location prediction technology can infer the next location of a user based on historical trajectory and provide flexible services for users, the latter of which is a current concern for organizations [6]–[8]. Previous studies have shown that 93% of user behavior is predictable [9]. To date, this technology has been widely used in social security [10], [11], intelligent transportation [12]–[17], and location services [18]–[20].

As a classical statistical model, the first-order Markov chain (1-MC) has strong interpretability and is widely used in time-series data prediction. However, 1-MC assumes that the location at the next moment is only related to the current location, which significantly limits the predictive performance of the model [21], [22]. For this reason, Sha *et al.* [23] proposed a multi-order Markov chain ($k$-MC) based on 1-MC. The $k$-MC assumes that the location at the next moment is related to the previous $k$ locations but is prone to problems related to dimensionality; i.e., its state space explodes with an increase in $k$, which renders $k$-MC less practical in the field of time-series predictions. In addition to the MC model, the hidden Markov model (HMM) [24], [25] and association rule mining algorithms [26], [27] can also be used for time-series location prediction, but are still not suitable for predicting long-term time-series data. To solve the long-term dependence problem, previous studies have applied deep learning models to time-series data prediction, such as the recurrent neural

The associate editor coordinating the review of this manuscript and approving it for publication was Juan A. Lara.

network (RNN) [28], long short-term memory (LSTM) [29], and gated-recurrent-unit (GRU) models [30]. Compared with the classical statistical model, deep learning models achieve higher prediction accuracy; however, they are data-driven empirical models that are hampered by difficulties interpreting causal relationships in the model.

Therefore, we propose a hybrid Markov-LSTM model, which combines the advantages of the Markov and LSTM models to mine user movement patterns based on the user's transition probability (i.e., the transition probability is interpretable and describes the movement tendency of the user), as well as improve the performance of the location prediction model. This study makes several significant contributions, which are summarized as follows:

(1) A new multi-step Markov transition probability matrix is presented, which divides the multi-order Markov model into multiple first-order models and solves the shortcomings of the multi-order Markov model in the dimension disaster.

(2) The prediction results of the multiple first-order Markov models are combined based on the advantages of the LSTM for predicting long-sequence data. This improves the practicality of the multi-order Markov model for location prediction.

(3) The performance of the Markov-LSTM model is evaluated using real indoor trajectories. The results demonstrate the advantages of our approach compared with five baseline methods.

The remainder of this study is organized as follows. In Section II, we review current literature on location prediction models based on trajectories. The basic definitions and problems are described in Section III. In Section IV, we propose a new methodological framework for destination prediction. The performance of this method and those proposed in previous studies are compared using real indoor Wi-Fi positioning data. These results are presented in Section V. In Section VI, we summarize the study and provide suggestions for future research.

## II. BACKGROUND

Existing location prediction methods can be divided into two approximate types: group-based and individual-based prediction models.

Group-based models consider that movement behavior ‘‘follows the crowd’’ to a certain degree and use the history trajectories of other users to predict user next location. These models are predominantly used to mine similarity behaviors from group users. For example, Morzy [27] used an improved apriori algorithm that uses association rules to predict the next location of user; Ang *et al.* [31] utilized a Markov chain to convert location sequences into transition probabilities for location prediction; Qiang *et al.* [32] presented a spatiotemporal RNN (ST-RNN) based on RNN [33] to model the location of group users; and Ying *et al.* [34] presented a geographic–temporal–semantic-based location prediction model to predict the next location of group users. Unlike single-object models, group-based models can mine the movement patterns of group users in certain scenarios [35]. In addition, group-based models do not require long-term trajectories of individual users. However, there are several deficiencies to these models. Group-based models build a model for all users, ignoring the existence of similarity subgroups. Therefore, some models only obtain the movement trajectories for people that are somehow related to the user. Zhang *et al.* [36] found a strong correlation between the calling patterns and co-cell patterns of users. Based on the results, they presented the NextCell model, which aims to enhance location prediction by harnessing the social interplay revealed by cellular call records. Moreover, Wen *et al.* [37] presented a fallback social-temporal-hierarchic Markov model (FSTHM), which used modified cross-sample entropy to quantify similarities between an individual and their peers to enhance predictive performance. Furthermore, Peixiao *et al.* [38] proposed a location prediction framework based on the similarity of location sequences.

Conversely, individual-based models consider that the movement behavior of each individual is independent; thus, they use only the movement history of the specific user to predict their next location. Individual-based models are predominantly used to mine the periodic behavior of individual users. For example, Lee *et al.* [39] presented a spatiotemporal-periodic (STP) pattern to capture the periodic behavior of an individual. Then, using an association rule algorithm to mine periodic patterns in the STP pattern, Vu *et al.* [40], [41] proposed a novel framework, named as Jyotish, to obtain the periodic movement of people based on Wi-Fi/Bluetooth positioning data. Bayesian classifiers and support vector machines were utilized to predict the next most likely location. Minh Tri Do *et al.* [42] redefined the location prediction problem from a new perspective and presented a probabilistic kernel method to determine the dependence between the user location and multivariate context variables from sparse data. Moreover, Wu *et al.* [43] presented a spatial-temporal-semantic neural network algorithm (STS-LSTM) for location prediction and Zhang *et al.* [12] combined the respective advantages of support vector regression and deep learning to present a novel data embedding and ensemble learning method. Furthermore, Zhou *et al.* [44] defined a novel Markov chain via Markov transition matrix multiplication and proposed the DestPD model.

However, the existing models suffer from certain deficiencies. First, group-based models require a large number of user trajectories and the prediction accuracy is relatively low. Second, individual-based models have better predictive performance but often require a significant amount of personal information. Finally, previous research has focused on location prediction in outdoor environments, with relatively few studies on indoor environments. Therefore, in this study, we develop a novel indoor location prediction model for an individual user, termed the Markov-LSTM. Compared with existing models, the proposed model only requires the trajectory of the user and combines the advantages of Markov and LSTM methods to improve the prediction performance.

## III. PRELIMINARIES AND PROBLEM DEFINITIONS

*Definition 1 (Trajectory):* A trajectory, $traj = \{pt_i\}_{i=1}^{n}$, is an ordered sequence of points for $pt_i = (id, t_i, x_i, y_i, f_i)$, where $id$ is a unique user identifier, $t_i$ is the time that $pt_i$ was collected, and $(x_i, y_i, f_i)$ corresponds to the longitude, latitude, and floor, respectively, of the user at time $t_i$.

*Definition 2 (Stay Point):* In general, a stay point or points, $sp^{id} = (x, y, f, arrT, levT)$ represent a geographic region in which the user remained over a certain time interval, where $id$ is the unique user identifier, $(x, y, f)$ corresponds to the average $x$, average $y$, and floor, respectively, on which the user stayed, and $(arrT, levT)$ represents the arrival and departure times, respectively, of the user in the geographic region [45]. As shown in Fig. 1a, the stay point of user $u$ is expressed as $sp^u = \left(\sum_{i=5}^{8} pt_i.x/4, \sum_{i=5}^{8} pt_i.y/4, 3, pt_5.t, pt_8.t\right)$.

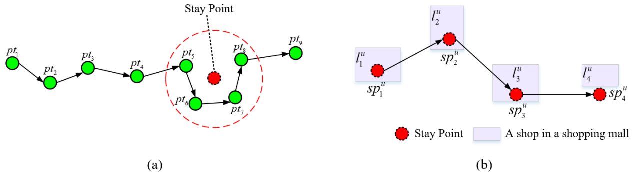

**FIGURE 1.** Basic definitions used in the prediction model: (a) movement of a user on the third floor and (b) the location sequence of a user in the indoor space.

*Definition 3 (Location Set):* The location set, $lset = \{l_i\}_{i=1}^{N}$, represents the set of regions in a specific application. The application employed in this study is shops in a mall, where $l_i = (lid, shape_i, f_i)$, $lid$ represents the unique identifier of shop $l_i$, $shape_i$ represents the limited area of shop $l_i$, $f_i$ represents the floor identifier where shop $l_i$ is located, and $N$ represents the number of shops in the mall.

*Definition 4 (Location Sequence):* A location sequence, $locSeq^{id} = \{l_i^{id}\}_{i=1}^{m}$, is an ordered sequence of locations visited by the user, where $l_i^{id}$ represents the shop visited at the stay point, $sp_i^{id}$. As shown in Fig. 1b, $locSeq^u = \left\{l_1^u, l_2^u, l_3^u, l_4^u\right\}$ represents the location sequence of user $u$.

The main object of this study is to analyze the location sequence, $locSeq^{id}$, of individual users as a method of determining the behavior patterns and living habits of individual users from their historical location sequences, which could aid future location prediction for that user. Taking user $u$ as an example, their location is defined in (1) and (2):

$$\hat{\boldsymbol{y}}^{res} = \mathcal{M}^u \leftarrow \left\{l_i^u\right\}_{i=1}^{m} \tag{1}$$

$$l_{m+1}^u = l_{\text{argmax}(\hat{\boldsymbol{y}}^{res})} \tag{2}$$

where $\left\{l_i^u\right\}_{i=1}^{m}$ represents the recent location sequence of user $u$; $\mathcal{M}^u$ represents the established prediction model based on the historical location sequence of user $u$; $\hat{\boldsymbol{y}}^{res} = \{\hat{y}_1^{res}, \hat{y}_2^{res}, \ldots, \hat{y}_i^{res}, \ldots, \hat{y}_N^{res}\}^T$ represents the prediction result of model $\mathcal{M}^u$, where $\hat{y}_i^{res}$ represents the probability that the user's next visit location is $l_i$; $l_{m+1}^u$ represents the result of the final prediction by the model; and argmax is a function that finds the index of the maximum value in $\hat{\boldsymbol{y}}^{res}$.

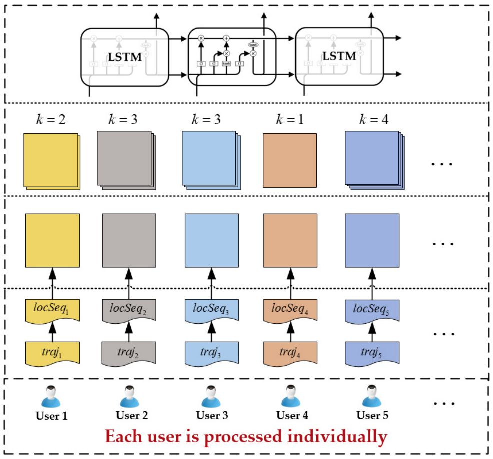

**FIGURE 2.** Schematic of the overall Markov-LSTM model process.

## IV. METHODOLOGY

In this section, we describe the proposed hybrid model for indoor location prediction, whose structure is presented in Fig. 2. Based on the bottom-up design principle, our method is divided into four phases: location sequence detection, multi-step transition probability matrix definition, adjacent location selection, and fusion multiple Markov chains, which are discussed in Sections IV.A and IV.B, respectively. First, considering the continuity of space, the trajectory is not suitable for direct input into the prediction model. Therefore, we must convert the trajectory into a location sequence associated with a specific shop. Second, a novel multi-step Markov transition probability matrix is defined, which converts a higher-order Markov chain into multiple first-order Markov chains. Third, we select the most appropriate adjacent locations for each user. Finally, the LSTM model is used to integrate these first-order Markov chains to obtain the predicted results of the target user.

### A. LOCATION SEQUENCE DETECTION METHOD

Stay point identification is one of the important steps in location sequence conversion. When the user is staying, there is a greater probability of viewing the location service information [46]. In this study, we used the indoor-STDBSCAN algorithm to detect the stay point, $sp^{id}$, from the indoor individual trajectory. The Indoor-STDBSCAN algorithm [38] divides the indoor individual trajectory, $traj$, into $k$ disjoint clusters $\{C_1, C_2, \ldots, C_k\}$, where $k$ clusters have $k$ stay points. Indoor-STDBSCAN, which is an improved version of the DBSCAN algorithm [47], redefines the spatiotemporal neighborhood of the indoor space based on DBSCAN. The spatiotemporal neighborhood of the trajectory point, $pt_i$, can be defined using the following expression:

$$\begin{aligned} N_{\epsilon_1,\epsilon_2}(pt_i) = \{&pt_j \in traj | sd(pt_j, pt_i) \\ &\leq \epsilon_1 \; and \; td\left(pt_j, pt_i\right) \\ &\leq \epsilon_2 \; and \; pt_j.f_j == pt_i.f_i\} \end{aligned} \tag{3}$$

where $sd$ is a function that calculates the spatial distance between $pt_i$ and $pt_j$, $td$ is used to calculate the time distance between $pt_i$ and $pt_j$, and $N_{\epsilon_1,\epsilon_2}(pt_i)$ represents a set of points contained in the spatiotemporal neighborhood.

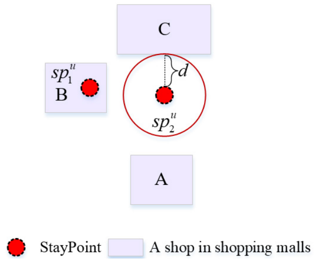

**FIGURE 3. Schematic showing the method of nearest-neighbor search.**

The stay point detected by the Indoor-STDBSCAN only contains spatial information, not semantic information. Therefore, we use the nearest-neighbor search to assign semantics to each stay point. As shown in Fig. 3. For a stay point $sp_1^u$ that is inside the shop, we use the intersection method to obtain the shop that user $u$ visited at stay point $sp_1^u$. For a stay point $sp_2^u$ that is outside the shop, the shop nearest to point $sp_2^u$ and the corresponding distance $d$ are determined. If $d$ is less than the distance threshold $\delta$, the nearest shop is that which the user visits at stay point $sp_2^u$.

## B. HYBRID MARKOV-LSTM MODEL

### 1) DEFINITION OF THE MULTI-STEP TRANSITION PROBABILITY MATRIX

The $k$-MC is a classic statistical model that describes the movement of a user between locations from a transition probability perspective. The k-MC treats each location in the user location sequence as a random variable, using the joint probability to predict the location of the user's next visit. Taking user $u$ as an example, a location sequence, $\left\{l_i^u\right\}_{i=1}^{m}$, of length $m$ can be expressed as a random variable, $\left\{L_i^u\right\}_{i=1}^{m}$, of length $m$. Moreover, the $k$-order probability transition matrix of user $u$, $\boldsymbol{Y}^u \in \mathbb{R}^{k\times N\times N}$, can be expressed as $\mathrm{P}(L_{m+1}^u|L_m^u, L_{m-1}^u, \ldots, L_{m-k+1}^u)$. With an increase in $k$, the transition probability matrix, $\boldsymbol{Y}^u$**,** of user $u$ increases rapidly, which renders $k$-MC less practical for location prediction. For this reason, we propose a novel $k$-step Markov chain, $\mathrm{MC}^{(k)}$.

*Definition 5 (1-Step Transition Probability Matrix):* The 1-step transition probability matrix, $\boldsymbol{Y}^{u(1)}$, of user $u$ is equivalent to the 1-order transition probability matrix, $\boldsymbol{Y}_{ij}^{u(1)}$, which represents the probability that user $u$ moves from location $l_i$ through one step to location $l_j$. $\boldsymbol{Y}_{ij}^{u(1)}$ is defined by the following expression:

$$\boldsymbol{Y}_{ij}^{u(1)} = \frac{\sum_{p=1}^{m-1} |\{l_p^u{=}l_i \cap l_{p+1}^u{=}l_j\}|}{\sum_{p=1}^{m} |\{l_p^u{=}l_i\}|} \quad \begin{matrix} \boldsymbol{Y}^{u(1)} \in \mathbb{R}^{N\times N} \\ l_p^u \in locSeq^u \end{matrix} \tag{4}$$

where $locSeq^u$ represents the location sequence, $\left\{l_i^u\right\}_{i=1}^{m}$, of user $u$, $\sum_{p=1}^{m-1} |\{l_p^u = l_i \cap l_{p+1}^u = l_j\}|$ represents the distance that user $u$ moves from location $l_i$ through one step to location $l_j$, $\sum_{p=1}^{m} |\{l_p^u = l_i\}|$ represents the total distance that user $u$ moves from location $l_i$ through one step to other locations, and $N$ represents the total number of shops in the mall.

*Definition 6 (**k**-Step Transition Probability Matrix):*. The $k$-step transition probability matrix, $\boldsymbol{Y}^{u(k)}$, of user $u$ is a $N \times N$ matrix, $\hat{\boldsymbol{y}}^{u(l_i\to *:k)} = \boldsymbol{Y}_{i\to *}^{u(k)}$, which represents the probability that user $u$ moves from location $l_i$ through $k$ steps to other locations. The definitions of $\boldsymbol{Y}^{u(k)}$ and $\hat{\boldsymbol{y}}^{u(l_i\to *:k)}$ for user $u$ are expressed in (5) and (6), respectively:

$$\boldsymbol{Y}^{u(k)} = \mathrm{P}(L_{m+1}^u|L_{m-k+1}^u)\boldsymbol{Y}^{u(k)} \in \mathbb{R}^{N\times N} \tag{5}$$

$$\hat{\boldsymbol{y}}^{u(l_i\to *:k)} = \mathrm{P}(L_{m+1}^u|L_{m-k+1}^u = l_{m-k+1}^u)\hat{\boldsymbol{y}}^{u(l_i\to *:k)} \in \mathbb{R}^{1\times N} \tag{6}$$

where $\boldsymbol{Y}^{u(k)}$ can be directly obtained by $\boldsymbol{Y}^{u(1)}$; i.e. $\boldsymbol{Y}^{u(k)} = \left(\boldsymbol{Y}^{u(1)}\right)^k$, $L_{m-k+1}^u$ represents a random variable of user $u$, $L_{m-k+1}^u = l_{m-k+1}^u$ indicates that user $u$ determines to visit location $l_{m-k+1}^u$ at random variable $L_{m-k+1}^u$ ($l_{m-k+1}^u$ can be obtained in the position sequence $locSeq^u$), $\boldsymbol{Y}^{u(k)}$ describes the effect that cross-location has on the prediction results from another perspective, and $N$ represents the total number of shops in the mall.

The aim of the $\mathrm{MC}^{(k)}$ is to establish a transition probability matrix of the same size as the 1-MC transition probability matrix. Using this matrix, the $k$-MC can be decomposed into $k$ first-order Markov chains in order to avoid solving the joint probability of $k$-MC and reduce the dimensions of the transition probability matrix to a certain extent. To make this theoretical analysis more rigorous, we provide mathematical proof of the k-step transition probability matrix $\boldsymbol{Y}^{u(k)}$. If the location of user $u$ at the random variable $L_{m-k}^u$ is $l_{m-k}^u$, the probability that user $u$ moves from location $l_{m-k}^u$ to other locations through $k$ steps can be defined using the following expressions:

$$\begin{aligned} &\hat{\boldsymbol{y}}^{(l_{m-k}^u\to *:k+1)} \\ &\quad = \hat{\boldsymbol{y}}^{(l_{m-k}^u\to *:1)} * \mathrm{P}\left(L_{m-k+2}^u \,|\, L_{m-k+1}^u\right) \\ &\qquad * \mathrm{P}\left(L_{m-k+3}^u \,|\, L_{m-k+2}^u\right) * \ldots\ldots * \mathrm{P}\left(L_{m+1}^u \,|\, L_m^u\right) \end{aligned} \tag{7}$$

$$\begin{cases} \hat{\boldsymbol{y}}^{(l_{m-k}^u\to *:1)} = \mathrm{P}\left(L_{m-k+1}^u \,|\, L_{m-k}^u = l_{m-k}^u\right) \\ \mathrm{P}\left(L_{m-k+2}^u \,|\, L_{m-k+1}^u\right) = \boldsymbol{Y}^{u(1)} \\ \mathrm{P}\left(L_{m-k+3}^u \,|\, L_{m-k+2}^u\right) = \boldsymbol{Y}^{u(1)} \\ \qquad\vdots \\ \qquad\vdots \\ \qquad \mathrm{P}\left(L_{m+1}^u \,|\, L_m^u\right) = \boldsymbol{Y}^{u(1)} \end{cases} \tag{8}$$

where $\mathrm{P}\left(L_{m-k+2}^u \,|\, L_{m-k+1}^u\right) * P\left(L_{m-k+3}^u \,|\, L_{m-k+2}^u\right) * \ldots\ldots * \mathrm{P}\left(L_{m+1}^u \,|\, L_m^u\right)$ represents the k-step transition probability matrix $\boldsymbol{Y}^{u(k)}$; i.e., $\boldsymbol{Y}^{u(1)} * \boldsymbol{Y}^{u(1)} * \ldots \boldsymbol{Y}^{u(1)}$. As matrix multiplication satisfies the associative law, $\boldsymbol{Y}^{u(1)} * \boldsymbol{Y}^{u(1)} * \ldots \boldsymbol{Y}^{u(1)}$ can be expressed as $\left(\boldsymbol{Y}^{u(1)}\right)^k$, i.e., $\boldsymbol{Y}^{u(k)} = \left(\mathbf{Y}^{u(1)}\right)^k$.

### 2) SELECTION OF THE BEST ADJACENT LOCATIONS

Similar to the $k$-MC, the Markov-LSTM model must also determine the hyper-parameter, $k$; i.e., the number of

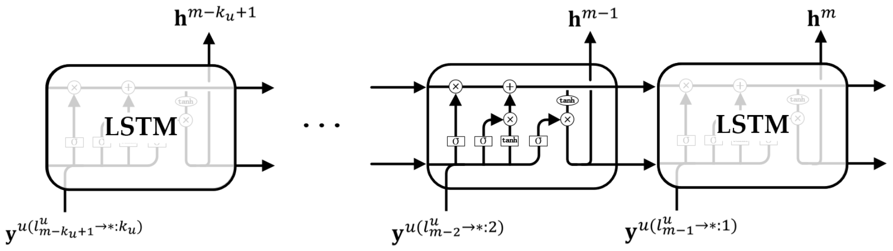
$\mathbf{h}^{m-k_u+1}$
$\mathbf{h}^{m-1}$
$\mathbf{h}^{m}$
LSTM
σ
tanh
LSTM
$\mathbf{y}^{u(l^u_{m-k_u+1}\to *:k_u)}$
$\mathbf{y}^{u(l^u_{m-2}\to *:2)}$
$\mathbf{y}^{u(l^u_{m-1}\to *:1)}$

**FIGURE 4. Fusion of multiple Markov models.**

locations that the prediction result depends on. The appropriateness of the parameters has a substantial influence on the prediction performance of the model. The $k$ value is primarily employed to determine the number of adjacent locations. If the $k$ value is too small, the model corresponds to a first-order Markov chain that reduces the prediction performance. If the $k$ value is too large, the model becomes more complex and overfitting is possible. Considering that the selection of the $k$ value significantly influences the prediction performance, this value is typically determined using cross-validation, which selects the $k$ value that minimizes the model prediction error [15], [48]. In this study, each user is an independent individual; therefore, we select the optimal $k$ value for different users. Taking user $u$ with a $k$ value of $k_u$ as an example, when $k_u > 1$, the $k$-MC can be decomposed based on the following expressions:

$$\begin{cases} \hat{\boldsymbol{y}}^{u(l^u_m\to *:1)} = \mathrm{P}\left(L^u_{m+1} \mid L^u_m = l^u_m\right) \\ \hat{\boldsymbol{y}}^{u(l^u_{m-1}\to *:2)} = \mathrm{P}\left(L^u_{m+1} \mid L^u_{m-1} = l^u_{m-1}\right) \\ \qquad\vdots \\ \hat{\boldsymbol{y}}^{u(l^u_{m-k_u+1}\to *:k_u)} = \mathrm{P}\left(L^u_{m+1} \mid L^u_{m-k_u+1} = l^u_{m-k_u+1}\right) \end{cases} \tag{9}$$

where $\left\{\hat{\boldsymbol{y}}^{u(l^u_{m-i+1}\to *:i)}\right\}_{i=1}^{k_u}$ represents the prediction results of multiple first-order Markov models for user $u$.

### 3) FUSION OF MULTIPLE MARKOV MODELS

For each user, $u$, we establish $k_u$ first-order Markov models. Each model, however, has a limited prediction ability for the next position. Therefore, this study combines $k_u$ first-order Markov models to ensure good location prediction performance. Considering the order of the $k_u$ first-order Markov model prediction results; i.e. $\left\{\hat{\boldsymbol{y}}^{u(l^u_{m-i+1}\to *:i)}\right\}_{i=1}^{k}$, we use the LSTM model to merge the $k_u$ results. Improvements in model prediction performance can be considered from two aspects: (1) from a Markov model perspective, the multi-step transition probability matrix allows the use of multiple 1-MCs to achieve $k$-MC predictive performance without problems regarding dimensional disasters; (2) from an LSTM model perspective, our model does not directly mine the location pattern from the simple identification sequence but rather mines the location pattern from the transition probability that contains more of the user's movement tendencies.

As an extension of the RNN model, the LSTM model has a unique cell that effectively controls the rate of information accumulation by introducing gate mechanisms (i.e., input gate, forgetting gate, and output gate) and selectively forgetting certain historical accumulation information. As shown in Fig. 4, the outputs of $k_u$ first-order Markov models are merged in turn using the input gate, forgetting gate, and output gate. This fusion method not only integrates the independent influence of multiple outputs on the prediction results but also determines the interaction between multiple outputs. The fusion process for user $u$ can be expressed with the following equations:

$$\begin{cases} \boldsymbol{f}^m = \sigma(\boldsymbol{W}_{hf}\boldsymbol{h}^{m-1} + \boldsymbol{W}_{yf}\boldsymbol{y}^{u(l^u_m\to *:1)} + \boldsymbol{b}_f) \\ \boldsymbol{i}^m = \sigma(\boldsymbol{W}_{hi}\boldsymbol{h}^{m-1} + \boldsymbol{W}_{yi}\boldsymbol{y}^{u(l^u_m\to *:1)} + \boldsymbol{b}_i) \\ \boldsymbol{a}^m = tanh(\boldsymbol{W}_{ha}\boldsymbol{h}^{m-1} + \boldsymbol{W}_{ya}\boldsymbol{y}^{u(l^u_m\to *:1)} + \boldsymbol{b}_a) \\ \qquad \boldsymbol{C}^m = \boldsymbol{C}^{m-1} \odot \boldsymbol{f}^m + \boldsymbol{i}^m \odot \boldsymbol{a}^m \\ \boldsymbol{o}^m = \sigma(\boldsymbol{W}_{ho}\boldsymbol{h}^{m-1} + \boldsymbol{W}_{oy}\boldsymbol{y}^{u(l^u_m\to *:1)} + \boldsymbol{b}_o) \\ \qquad \boldsymbol{h}^m = \boldsymbol{o}^m \odot \tanh\left(\boldsymbol{C}^m\right) \\ \qquad \hat{\boldsymbol{y}}^m = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^m + \boldsymbol{b}_h\right) \end{cases} \tag{10}$$

In this algorithm, $\boldsymbol{f}^m$, $\boldsymbol{i}^m$, $\boldsymbol{C}^m$, and $\boldsymbol{o}^m$ represent the forgetting gate, input gate, control unit, and output gate, respectively, $\boldsymbol{h}^{m-1}$ represents the hidden unit of the correlation between the outputs of multiple Markov models, $\boldsymbol{W}_{hf}$, $\boldsymbol{W}_{yf}$, $\boldsymbol{W}_{hi}$, $\boldsymbol{W}_{yi}$, $\boldsymbol{W}_{ha}$, $\boldsymbol{W}_{ya}$, $\boldsymbol{W}_{ho}$, $\boldsymbol{W}_{oy}$, and $\boldsymbol{W}_h$ represent the weight matrices, $\odot$ represents the Hadamard operation, $\hat{\boldsymbol{y}}^m$ represents the output of the Markov-LSTM model; i.e., $\hat{\boldsymbol{y}}^{res}$ in the problem definition, and $\sigma$ represents the sigmoid activation function.

Our Markov-LSTM can be trained to predict $\boldsymbol{y}^m$ by merging multiple Markov outputs in order to minimize the cross-entropy loss between the predicted and true locations of a user. This loss function can be defined by the following expression:

$$\mathcal{L}\left(\theta\right) = -\sum_{j=1}^{N} y^m_j * log(\hat{y}^m_j) \tag{11}$$

where $\theta$ represents all learnable parameters; i.e., all $\boldsymbol{W}$ and $\boldsymbol{b}$ parameters, in the Markov-LSTM model; $N$ represents the total number of locations; i.e., the number of shops; $\hat{y}^m_j$ represents the output of the model; and $y^m_j$ represents the expected output (true value) of the model.

#### 4) ALGORITHMS AND OPTIMIZATION

In this study, the location sequence of user $u$, $locSeq^u$, is divided into three parts: historical samples, training samples, and test samples. The historical location sequence is used to construct the k-step transition probability matrix, $\boldsymbol{Y}^{u(k)}$, of user $u$; the training location sequence is used to train the $\theta$ parameter of model $\mathcal{M}^u$; and the test location sequence is used to test the prediction performance of model $\mathcal{M}^u$. Algorithm 1 shows the $\mathcal{M}^u$ training process.

**Algorithm 1** Markov-LSTM Training Process

```
Require: Individual trajectory: $traj = \{(u, t_i, x_i, y_i, f_i)\}_{i=1}^{m}$
         Hyper-parameters of Indoor-STDBSCAN:
           $\epsilon_1, \epsilon_2, MinPts$
         Distance threshold: $\delta$
         Length of adjacency locations: $k_u$
Construct $locSeq^u$ based on $\epsilon_1, \epsilon_2, MinPts$, and $\delta$
//construct first-step transition probability matrix
Divide the $locSeq^u$ into $his\_locSeq^u$, $tr\_locSeq^u$, and $te\_locSeq^u$
Construct $\boldsymbol{Y}^{u(1)}$ based on (4) with $his\_locSeq^u$
//construct training instances
$\mathcal{D} \leftarrow \emptyset$
For next $i \in [k_u, k_u + 1, \ldots, tr\_locSeq^u]$ do
    Construct $\boldsymbol{Y}^{u(2)}, \boldsymbol{Y}^{u(3)}, \ldots, \boldsymbol{Y}^{u(k_u)}$ by $\boldsymbol{Y}^{u(1)}$
    Obtain index of the previous $k_u$ locations of
      $tr\_locSeq^u[i]$: $ix_i, ix_{i+1}, \ldots, ix_{k_u}$
    Put a training instance
      $(\{\boldsymbol{Y}^{u(1)}[ix_i], \ldots, \boldsymbol{Y}^{u(k_u)}[ix_{k_u}]\}s, tr\_locSeq^u[i])$
      into $\mathcal{D}$
//train the model
Initialize the parameters $\theta$
Repeat
  Randomly select a batch of instances $\mathcal{D}_b$ from $\mathcal{D}$
  Find $\theta$ by minimizing (11) with $\mathcal{D}_b$
Until stopping criteria is met
Output the learned Markov-LSTM model $\mathcal{M}^u$
```

#### 5) MODEL COMPLEXITY ANALYSIS

Definition of the $k$-step Markov model is a key step in Markov-LSTM model. In this section, we analyze the advantages of the $k$-step Markov model from two perspectives: space complexity and computational complexity.

From the perspective of space complexity, the number of elements in the $k$-step transition probability matrix is $k * N * N$; however, the number of elements in the $k$-step transition probability matrix is $N * N$. Compared with the $k$-order Markov transition probability matrix, the storage space required for the $k$-step Markov transition probability matrix is significantly reduced. Especially with an increase of $k$, the advantages of the $k$-step Markov model become increasingly significant.

From the perspective of computational complexity, according to (7) and (8), the $k$-step Markov model only needs to calculate the first-order Markov transition probability matrix. Compared with the $k$-order Markov model, the $k$-order Markov transition probability matrix has higher computational complexity. Especially with an increase of k, the computational complexity of the k-order Markov transition probability matrix becomes increasingly large [49]. However, the $k$-step Markov transition probability matrix does not increase with an increase of $k$.

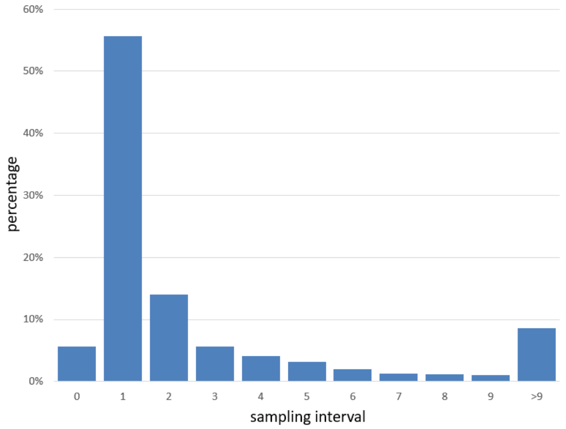

**FIGURE 5.** Sampling interval distribution of the trajectory data.

**TABLE 1.** Samples of user trajectory data.

| User ID | Date time | $X$ (m) | $Y$ (m) | Floor ID |
|---|---|---|---|---|
| 0000DE*** | 12/23/2017 10:46:45 | 130219*** | 43904*** | 1 |
| 0000DE*** | 12/23/2017 10:46:57 | 130219*** | 43903*** | 1 |
| 0000DE*** | 12/23/2017 10:47:05 | 130219*** | 43904*** | 1 |
| …… | …… | …… | …… | …… |
| 0000DE*** | 02/01/2017 19:20:33 | 130219*** | 43904*** | 4 |
| 0000DE*** | 02/01/2017 19:20:45 | 130219*** | 43904*** | 4 |

## V. EXPERIMENTAL RESULTS AND ANALYSIS

### A. DATA PREPARATION

#### 1) DATASETS

The experimental data predominantly consisted of Wi-Fi positioning information on 50 users and shops data for a shopping mall in Jinan City, China. The indoor Wi-Fi data was provided by Shanghai Palmap Science & Technology Company Limited (http://www.palmap.cn/) and collected using fingerprint positioning technology. The data covered the eight floors of the shopping mall from December 20, 2017, to February 1, 2018. The positioning accuracy was approximately 3 m. Fig. 5 shows the data sampling interval. Trajectory points with a sampling interval of 1–5 s accounted for more than 70% of the collected data points. There were a total of 11,677,438 trajectory points and each user had an average of 200,000 trajectory points. As shown in Table 1, the data

**TABLE 2.** Sample shops in the shopping mall.

| Shop ID | Shape | Name | Floor ID |
|---|---|---|---|
| 1 | Polygon | *** | 3 |
| 2 | Polygon | *** | 3 |
| 3 | Polygon | *** | 5 |
| …… | …… | …… | …… |
| 488 | Polygon | *** | 1 |
| 489 | Polygon | *** | 2 |

**TABLE 3.** Specifications of the experimental environment.

| Hardware | Setting |
|---|---|
| Operating system | Windows 10 |
| Central processing unit (CPU) | Intel$^{(R)}$ Core$^{TM}$ i7-8700 CPU @ 3.20 GHz |
| Random access memory (RAM) | 32.00 GB |
| Hard drive capacity | 2048 GB hard disk drive (HDD) |
| **Software** | **Setting** |
| Programming language | Python 3.6 |
| Platform/framework | PyTorch 0.4.0 |

field included the user unique identifier, the record upload time, the user's XY-coordinates, and the unique identifier of the floor. As shown in Table 2, there are 489 shops in the mall, each with an average size of approximately 40 m$^2$. Data for each shop included its unique ID, its shape (a polygon consisting of a sequence of coordinates), its name, and the floor ID.

### 2) DATA PREPROCESSING

The original trajectory data for the indoor users were collected through Wi-Fi positioning. Due to the unstable mobile terminal signal and the artificial shutdown of Wi-Fi signal, abnormal, erroneous, and invalid data were easily generated. There were three types of noise in our data set:

(1) Abnormal coordinate points. If the trajectory point fell outside the study area, it was treated as an abnormal coordinate trajectory point.

(2) Abnormal time points. If the sampling interval of two adjacent trajectory points was 0 s, it was considered an abnormal time trajectory point.

(3) Abnormal floor points. If a trajectory point was not in the study area or jumped between different floors in a short time period, it was considered an abnormal floor point.

### 3) TESTBED AND TEST DEVICE

Table 3 lists the experimental environment from two aspects: hardware and software.

## B. EVALUATION METRICS AND COMPARATIVE METHODS

### 1) EVALUATION METRICS

In this study, we treat location prediction as a classification problem using $Accuracy@X$, $Precision@X$, $Recall@X$, and $F1-Measure@X$ (top $X$ locations) as quantitative indicators of the evaluation model [50]. $Accuracy@X$ evaluates the predictive performance of the model from the perspective of binary classification; i.e., whether the top K predicted shops were actually visited. $Precision@X$, $Recall@X$, and $F1 - Measure@X$ use macro-averaging to evaluate model performance using multiple classifications. To fully test the prediction performance of the Markov-LSTM model, this study used the top 1, 3, and 5 locations to test the prediction ability of the model; i.e. $X \in \{1, 3, 5\}$. The $Accuracy@X$, $Precision@X$, $Recall@X$, and $F1-Measure@X$ are defined in (12), (13), (14), and (15), respectively:

$$Accuracy@X = \frac{number\ of\ samples\ correctly\ predicted}{total\ number\ of\ test\ samples} \tag{12}$$

$$Precision@X = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i} \tag{13}$$

$$Recall@X = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i} \tag{14}$$

$$F1-Measure@X = \frac{2 \times Precision@X \times Recall@X}{Precision@X + Recall@X} \tag{15}$$

where $N$ represents the total number of locations, that is, the total number of shops, $TP_i$ represents the number of samples for which the model correctly predicts that a user will visit shop $l_i$, and $FN_i$ represents the number of samples for which the model incorrectly predicts that a user will not visit shop $l_i$. When the predicted shop ID is equal to the shop ID of an actual visit, the prediction is considered correct, and vice versa.

### 2) COMPARATIVE METHODS

To comprehensively evaluate the performance of the Markov-LSTM model, we used five baseline methods for comparison:

**MC**[21]: A Markov chain (MC) is a statistical model used to describe discrete-time stochastic processes with Markov properties. In our experiment, we compared 1-MC, where 1-MC represents the first-order Markov chain.

**HMM**[24]: A hidden Markov model (HMM) is a statistical model used to describe a Markov process with implicit unknown parameters. The performance of the HMM model corresponds to the number of states. We fixed the number of states to one of 10, 15, 20. The three HMM variants were then compared; i.e., HMM-10, HMM-15, and HMM-20.

**RNN**[28]: A recurrent neural network (RNN) is a deep learning model that can determine temporal dependencies. Formally, the performance of the RNN model corresponds to the number of hidden states. In our experiment, the number of hidden states was fixed to one of {64, 128, 256} . Therefore, the three RNN variants were RNN-64, RNN-128, and RNN-256.

**LSTM**[29]: A long-short-term-memory network (LSTM) is a new type of RNN capable of learning long-term temporal dependencies. We conducted the experiments on three LSTM variants; i.e., LSTM-64, LSTM-128, and LSTM-256, which had identical settings to the RNN.

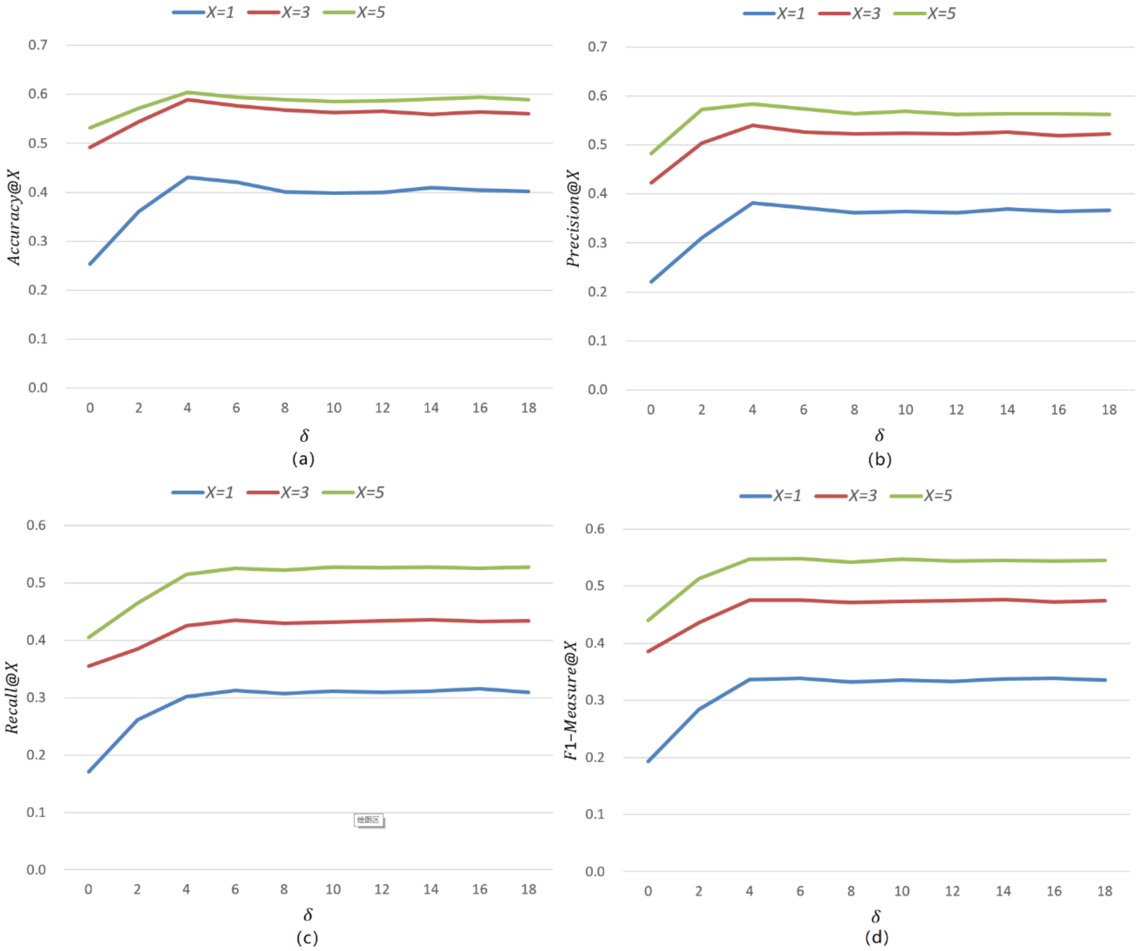

**FIGURE 6.** Impact of the distance threshold ($\delta$) on the prediction performance.

**GRU**[30]: A gated-recurrent-unit network (GRU) is a special RNN that can be used to determine long-term temporal dependencies. The GRU variants selected as the baselines were GRU-64, GRU-128, and GRU-256, which had identical settings to the RNN.

### C. VARIABLE ESTIMATION

The hyper-parameters of the Markov-LSTM model predominantly include the radius, $\epsilon_1$, time window, $\epsilon_2$, minimum number of points, *MinPts*, distance threshold, $\delta$, number of best adjacent locations, $k$, and parameters in the LSTM. The Indoor-STDBSCAN algorithm fixes $\epsilon_1$, $\epsilon_2$, and *MinPts* to 5 m, 7 min, and 100, respectively [38]. To further determine the $\delta$, $k$, and parameters in the LSTM, we used the control variable method to obtain the combination of parameter values. In the parameter calibration phase, the distance threshold, $\delta$, was first determined, followed by the optimal $k$ value based on the distance threshold. Finally, the LSTM parameter was adjusted to obtain the optimal model parameter combination.

#### 1) CALIBRATING THE DISTANCE THRESHOLD

The distance threshold, $\delta$, predominantly determines the influence of shop matching results on the prediction performance. If $\delta$ tends toward 0, the shop information only matches the stay point inside the shop. If $\delta$ tends toward infinity, any stay point will match the shop information. In this study, $\delta$ obtains the optimal parameter from $[0, 2, 4, \ldots, 18]$. Fig. 6 shows the effect of the distance threshold, $\delta$, on the prediction performance. When $X \in \{1, 3, 5\}$, $Accuracy@X$ and $Precision@X$ first showed an increasing trend followed by a decreasing trend that finally stabilized. $Recall@X$ AND $F1-Measure@X$ first showed an increasing trend then a stable trend. When $\delta > 6$M, the model prediction result did not change significantly because the indoor space was small. If $\delta$ is too large, $\delta$ will not act as a constraint. When $4 \leq \delta \leq 6$, the model exhibited better prediction performance. In this work, we fixed the distance threshold, $\delta$, to 4 m.

#### 2) CALIBRATING THE NUMBER OF BEST ADJACENT LOCATIONS

In the Markov-LSTM model, the number of best adjacent locations, $k$, plays an important role in the prediction process. We suggest that all users are independent individuals; thus, we select an optimal $k$ value for each individual. During the parameter calibration process, we set the range of $k$ to $[1, 2, \ldots, 30]$ and use cross-validation to obtain the optimal

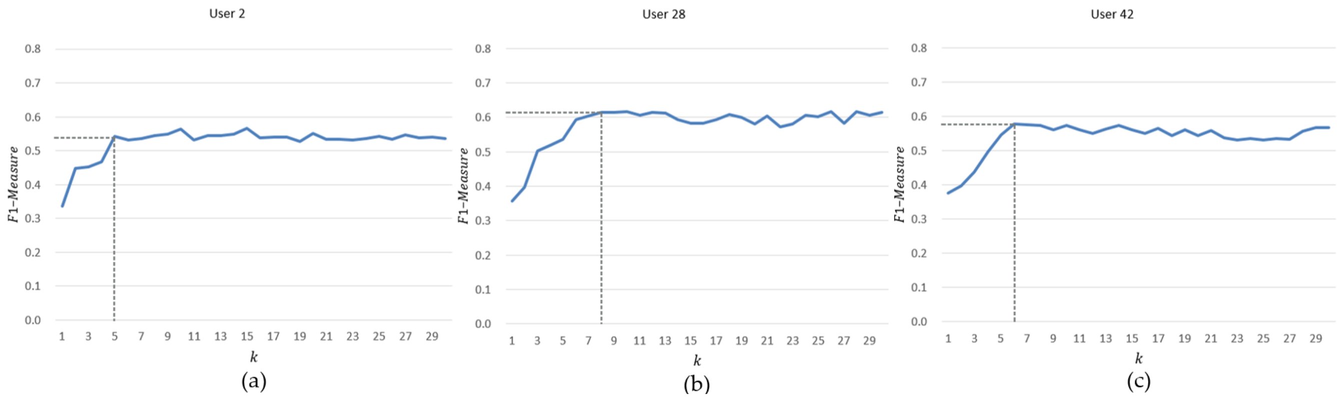

**FIGURE 7.** Impact of the number of best adjacent locations, $k$, on $F1 - Measure@1$ for different users: (a) User 2, (b) User 28, and (c) User 42.

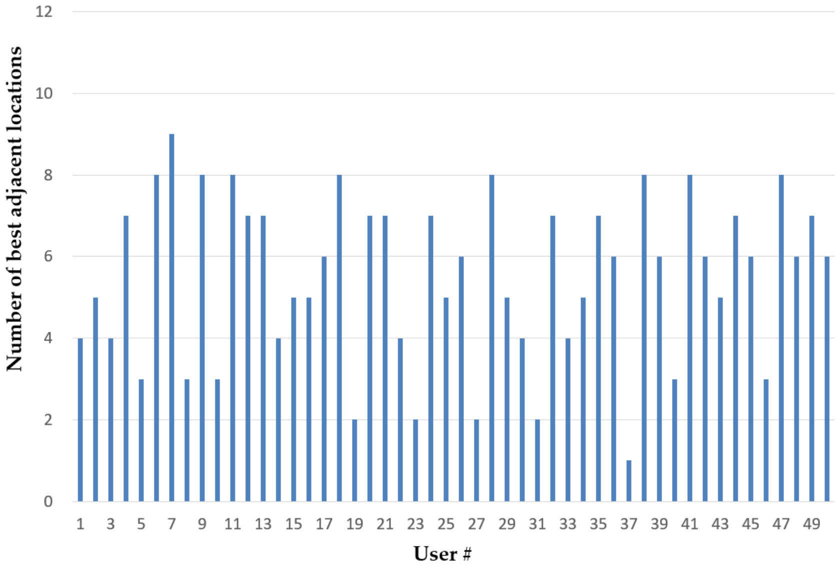

**FIGURE 8.** Number of best adjacent locations for each user.

combination of parameters for each user; this was performed to obtain the best $k$ value and optimal prediction performance. To illustrate the experimental results more simply, we randomly selected three users for $F1-Measure@1$ as examples with which to calibrate the hyper-parameters. These users were User 2, User 28, and User 42.

Fig. 7 shows the effect of hyper-parameter $k$ on the prediction model. The three users showed a consistent trend. As the $k$ value increased, the prediction performance exhibited a rapid increase. When the $k$ value reached a certain value, the predictive performance of the model began to stabilize. These results allowed us to obtain the optimal $k$ value for the three users (i.e., $k_{user2} = 5$, $k_{user28} = 8$, *and* $k_{user42} = 6$). Fig. 8 shows the different optimal $k$ values for each user, which reflects the fact that each user is an independent individual. These results also reflect the appropriateness of the method.

### 3) CALIBRATING THE LSTM PARAMETERS

In addition, we further validated the impact of LSTM parameters on model performance. In the LSTM, we calibrated the number of hidden states and set the range of hidden states to [32, 64, 128, 256, 512, 1024]. Fig. 9 shows the prediction results. As the hidden size increased, the model predictive performance first increased then became stable. When the hidden size was 128, the model exhibited better prediction performance.

### 4) MARKOV-LSTM MODEL PERFORMANCE

After determining the optimal combination of parameters, the model proposed in this study was further analyzed from the perspective of model prediction performance. Fig. 10 displays the results, which are summarized below.

(1) Comparing the four indicators for the dataset, the model prediction performance gradually increased with an increase in $X$. This was particularly true when $X = 3$, when $Accurary@3$, $Precision@3$, $Recall@3$, and $F1 - Measure@3$ reached 72.07%, 69.57%, 61.38%, and 65.22%, respectively. Compared with $X = 1$, the indicators improved by 6.4%, 6.53%, 4.4%, and 5.36, respectively. Compared with $X = 5$,

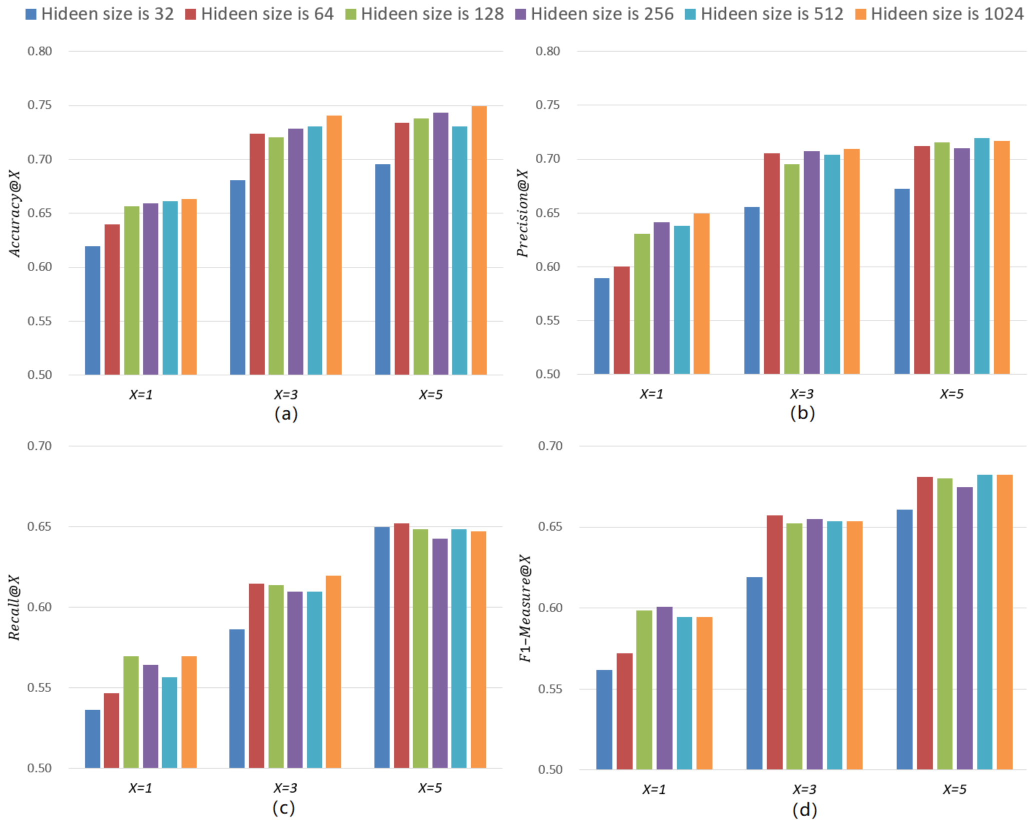

**FIGURE 9.** Impact of LSTM parameters on prediction performance: (a) location prediction accuracy, (b) location prediction precision, (b) location prediction recall, and (d) location prediction f1-measure.

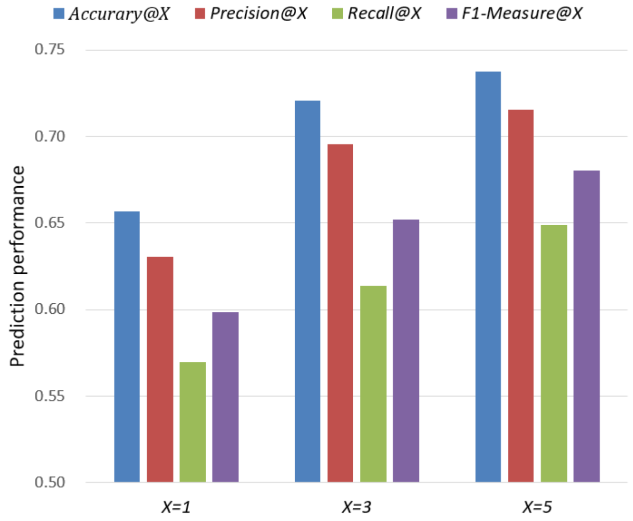

**FIGURE 10.** Location prediction performance of the Markov-LSTM model.

the indicators were only reduced by 1.7%, 1.96%, 3.49%, and 2.82%, respectively. When $X$ increased from 1 to 3, the performance of the model was greatly improved. However, when $X$ increased beyond 3, the performance of the model deteriorated. Thus, if the value of $X$ is low, the prediction performance of the model is low. If the value of $X$ is too high, the model prediction results would not have much value. The experiments determined that $X = 3$ is the most suitable value for this study.

(2) Compared to the $Accuracy@3$ value of the model, the $Precision@3$, $Recall@3$, and $F1 - Measure@3$ for the Markov-LSTM were reduced by 2.5%, 10.69%, and 6.85%, respectively. This is predominantly because the $Accuracy@3$, which is more suitable for a binary classifier, can be misleading. However, the $Precision@3$, $Recall@3$, and $F1 - Measure@3$ values suggest that location prediction is a multi-classification problem. As the test samples of each classification were unbalanced, there was a slight decrease in indicator values.

### 5) COMPARISON WITH BASELINE METHODS

In this section, the collected indoor trajectory data was used to test the prediction performance of the Markov-LSTM model with five existing baseline methods. These baseline methods can be approximately divided into two categories. The first category includes the MC and HMM methods, which are regarded as classic statistical prediction models. The second category includes the RNN, LSTM, and GRU methods, which are regarded as data-driven deep learning

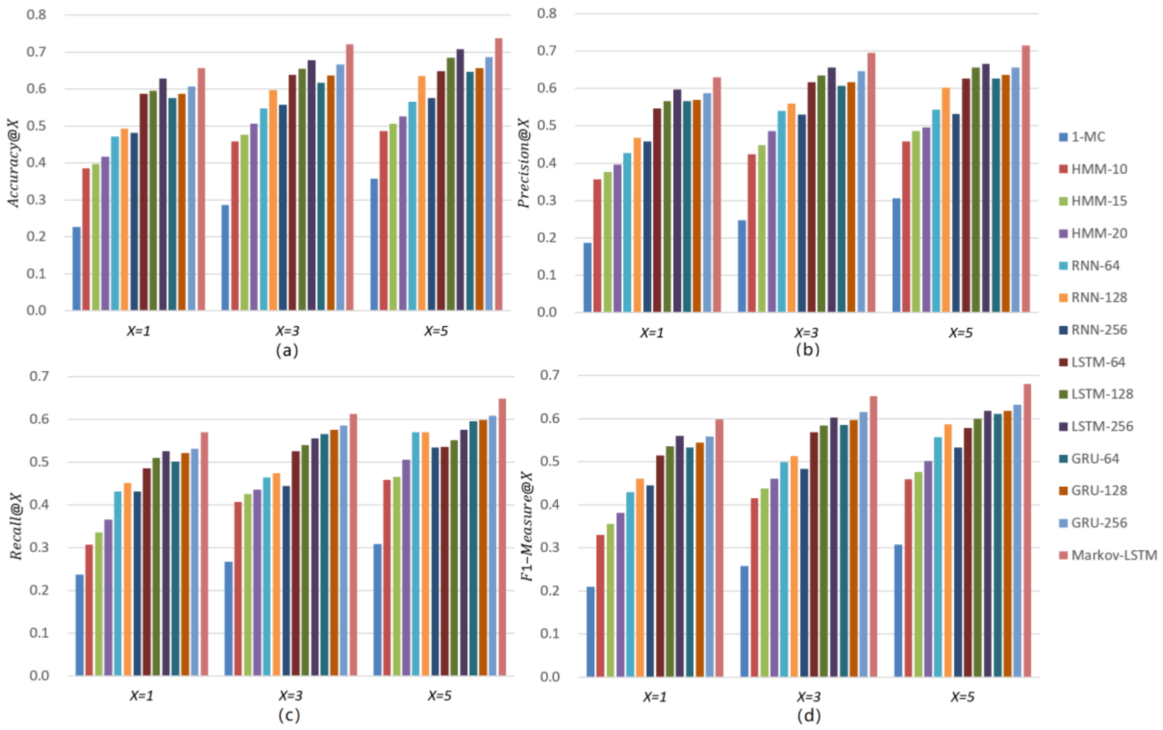

**FIGURE 11.** Comparisons of baseline methods and the Markov-LSTM model using the shopping mall dataset: (a) location prediction accuracy, (b) location prediction precision, (b) location prediction recall, and (d) location prediction f1-measure.

prediction models. The experimental results were analyzed using $Accuracy@X$, $Precision@X$, $Recall@X$, and $F1 - Measure@X$. Fig. 11 compares the prediction performance of the five models.

(1) Overall, the prediction performance of the second model type was significantly higher than that of the first type, whereas the prediction performance of the Markov-LSTM model was slightly higher than that of the second model type. If we take $X = 3$ as an example, the average $Accuracy@3$, $Precision@3$, $Recall@3$, and $F1 - Measure@3$ values of the 1-MC and HMM models were 39.64%, 36.71%, 35.21%, and 35.87%, respectively. Conversely, the average $Accuracy@3$, $Precision@3$, $Recall@3$, and $F1 - Measure@3$ values of the RNN, LSTM, and GRU models were 64.74%, 62.10%, 55.91%, and 58.84%, respectively. In comparison, the four indicators for the Markov-LSTM model were improved by 7.33%, 7.47%, 5.46%, and 6.38%, respectively.

(2) In more detail, the 1-MC model achieved poor prediction performance, with $Accuracy@3$, $Precision@3$, $Recall@3$, and $F1 - Measure@3$ values of 28.64%, 24.77%, 26.36%, and 25.54%, respectively. Conversely, corresponding values of 67.79%, 65.78%, 55.15%, and 55.99%, respectively, indicated good predictive performance for the LSTM model. Overall, the Markov-LSTM model significantly improved indoor location prediction performance by enhancing the $Accuracy@3$ by 6.29–43.43%, $Precision@3$ by 3.79–44.8%, $Recall@3$ by 9.23–35.02%, and $F1 - Measure@3$ by 13.80–39.68%.

## VI. CONCLUSION AND FUTURE WORK

Accurate and robust indoor location prediction plays an important role in indoor location services, particularly in the retail industry. For example, the ability to predict the next shop visited by users, as well as push information to the user on shops of interest, not only provides a personalized shopping experience to users but also boosts profits for retailers. Markov chains have been widely adopted for location prediction due to their strong interpretability; however, the $k$-MC is not suitable for predicting long sequences due to problems related to dimensionality. In this study, we proposed a novel hybrid Markov-LSTM model for indoor location prediction. First, a multi-step Markov transition matrix was defined, which decomposed a $k$-MC into multiple 1-MCs, solving the dimensional problem of the $k$-MC. Then, the LSTM model was introduced to merge multiple 1-MCs and improve model prediction performance. Experiments were conducted using real indoor trajectories from 50 users over 45 days to verify the predictive performance of the proposed model. First, we used the control variable method to obtain the optimal parameter combination of the Markov-LSTM model. When employing the optimal parameter combination, the model evaluation indicators $Accuracy@3$, $Precision@3$, $Recall@3$, and $F1 - Measure@3$ were 72.07%, 69.57%, 61.38%, and 65.22%, respectively. Then, we analyzed the predictive performance of the Markov-LSTM model using the test dataset. We conducted a comparison with five existing baseline methods, including the MC, HMM, RNN, LSTM, and GRU models. Compared with the existing methods, the Markov-LSTM model significantly improved indoor location prediction performance by enhancing $Accuracy@3$ by 6.29–43.43%, $Precision@3$ by 3.79–44.8%, $Recall@3$ by 9.23–35.02%, and $F1 - Measure@3$ by 13.80–39.68%. These results demonstrated the predictive performance of the Markov-LSTM model.

The hybrid Markov-LSTM model is a generalized prediction model that can be applied to more than simply indoor environments in future research. However, before its wider application, the following aspects require further study: (1) verification of the proposed model with a variety of data sources such as GPS trajectories, (2) comprehensive comparisons with other prediction models, and (3) integrate more factors to boost model robustness, thereby further improving the performance of location prediction.

## ACKNOWLEDGMENT

The authors would like to thank Shanghai Palmap Science & Technology Limited Company for providing essential indoor trajectory data support.

## REFERENCES

[1] N. Hajli and M. S. Featherman, ''Social commerce and new development in e-commerce technologies,'' *Int. J. Inf. Manage.*, vol. 37, no. 6, pp. 177–178, 2017, doi: 10.1016/j.ijinfomgt.2017.03.001.

[2] Y. Liu, ''Inferring gender and age of customers in shopping malls via indoor positioning data,'' *Environ. Planning B, Urban Anal. City Sci.*, Apr. 2019, doi: 10.1177/2399808319841910.

[3] W.-Q. Huang, C. Ding, S.-Y. Wang, and S. Hu, ''An efficient clustering mining algorithm for indoor moving target trajectory based on the improved AGNES,'' in *Proc. IEEE Trustcom/Bigdatase/Ispa*, Aug. 2015, pp. 1318–1323, doi: 10.1109/Trustcom.2015.524.

[4] S. Tomažič, D. Dovžan, and I. Škrjanc, ''Confidence-interval-fuzzy-model-based indoor localization,'' *IEEE Trans. Ind. Electron.*, vol. 66, no. 3, pp. 2015–2024, Mar. 2018, doi: 10.1109/TIE.2018.2840525.

[5] H. Li, L. Hua, L. Shou, C. Gang, and C. Ke, ''In search of indoor dense regions: An approach using indoor positioning data,'' *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1481–1495, Aug. 2018, doi: 10.1109/TKDE.2018.2799215.

[6] J. Xu, J. Zhao, R. Zhou, C. Liu, P. Zhao, and L. Zhao, ''Destination prediction a deep learning based approach,'' *IEEE Trans. Knowl. Data Eng.*, to be published, doi: 10.1109/TKDE.2019.2932984.

[7] D. Kong and F. Wu, ''HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction,'' in *Proc. IJCAI*, 2018, pp. 2341–2347.

[8] J. Zhao, J. Xu, R. Zhou, P. Zhao, C. Liu, and F. Zhu, ''On prediction of user destination by sub-trajectory understanding: A deep learning based approach,'' in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1413–1422.

[9] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, ''Limits of predictability in human mobility,'' *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010, doi: 10.1126/science.1177170.

[10] M. A. Tayebi, M. Ester, U. Glässer, and P. L. Brantingham, ''CRIMETRACER: Activity space based crime location prediction,'' in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2014, pp. 472–480.

[11] A. Bogomolov, B. Lepri, J. Staiano, N. Oliver, F. Pianesi, and A. Pentland, ''Once upon a crime: Towards crime prediction from demographics and mobile data,'' in *Proc. 16th Int. Conf. Multimodal Interact.*, 2014, pp. 427–434, doi: 10.1145/2663204.2663254.

[12] X. Zhang, Z. Zhao, Y. Zheng, and J. Li, ''Prediction of taxi destinations using a novel data embedding method and ensemble learning,'' *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2018.2888587.

[13] X. Li, M. Li, Y.-J. Gong, X.-L. Zhang, and J. Yin, ''T-DesP: Destination prediction based on big trajectory data,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2344–2354, Aug. 2016, doi: 10.1109/TITS.2016.2518685.

[14] C. Shifen, L. Feng, P. Peng, and W. Sheng, ''Multi-task and multi-view learning based on particle swarm optimization for short-term traffic forecasting,'' *Knowl.-Based Syst.*, vol. 180, pp. 116–132, Sep. 2019, doi: 10.1016/j.knosys.2019.05.023.

[15] S. Cheng, F. Lu, P. Peng, and S. Wu, ''Short-term traffic forecasting: An adaptive ST-KNN model that considers spatial heterogeneity,'' *Comput. Environ. Urban Syst.*, vol. 71, pp. 186–198, Sep. 2018.

[16] P. Cai, Y. Wang, G. Lu, C. Peng, C. Ding, and J. Sun, ''A spatiotemporal correlative k -nearest neighbor model for short-term traffic multistep forecasting,'' *Transp. Res. C, Emerg. Technol.*, vol. 62, no. 62, pp. 21–34, 2016.

[17] S. Cheng, F. Lu, P. Peng, and S. Wu, ''A spatiotemporal multi-view-based learning method for short-term traffic forecasting,'' *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 6, p. 218, 2018. [Online]. Available: http://www.mdpi.com/2220-9964/7/6/218.

[18] M. Ye, P. Yin, and W.-C. Lee, ''Location recommendation for location-based social networks,'' in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, 2010, pp. 458–461, doi: 10.1145/1869790.1869861.

[19] L. Kuang, ''A personalized QoS prediction approach for CPS service recommendation based on reputation and location-aware collaborative filtering,'' *Sensors*, vol. 18, no. 5, p. 1556, 2018, doi: 10.3390/s18051556.

[20] J. Bao, Y. Zheng, and M. F. Mokbel, ''Location-based and preference-aware recommendation using sparse geo-social networking data,'' in *Proc. ACM Int. Symp. Adv. Geogr. Inf. Syst.*, 2012, pp. 199–208, doi: 10.1145/2424321.2424348.

[21] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, ''Next place prediction using mobility Markov chains,'' in *Proc. 1st Workshop Meas., Privacy, Mobility (MPM)*, 2012, doi: 10.1145/2181196.2181199.

[22] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, ''Show me how you move and I will tell you who you are,'' *Trans. Data Privacy*, vol. 4, no. 2, pp. 103–126, 2011.

[23] W. Sha, Y. Zhu, M. Chen, and T. Huang, ''Statistical learning for anomaly detection in cloud server systems: A multi-order Markov chain framework,'' *IEEE Trans. Cloud Comput.*, vol. 6, no. 2, pp. 401–413, Apr./Jun. 2015.

[24] W. Mathew, R. Raposo, and B. Martins, ''Predicting future locations with hidden Markov models,'' in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, 2012, pp. 911–918, doi: 10.1145/2370216.2370421.

[25] J. A. Alvarez-Garcia, J. A. Ortega, L. Gonzalez-Abril, and F. Velasco, ''Trip destination prediction based on past GPS log using a hidden Markov model,'' *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8166–8171, 2010.

[26] T. I. Keles, M. Ozer, I. Toroslu, and P. Karagoz, ''Location prediction of mobile phone users using apriori-based sequence mining with multiple support,'' in *New Frontiers in Mining Complex Patterns* (Lecture Notes in Artificial Intelligence), vol. 8983. Heidelberg, Germany: Springer, 2014, pp. 179–193, doi: 10.1007/978-3-319-17876-9_12.

[27] M. Morzy, ''Prediction of moving object location based on frequent trajectories,'' in *Computer and Information Sciences—ISCIS*. Heidelberg, Germany: Springer, 2006, pp. 583–592, doi: 10.1007/11902140_62.

[28] A. Graves, ''Generating sequences with recurrent neural networks,'' 2013, *arXiv:1308.0850*. [Online]. Available: https://arxiv.org/abs/1308.0850

[29] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[30] F. Rui, Z. Zuo, and L. Li, ''Using LSTM and GRU neural network methods for traffic flow prediction,'' in *Proc. Youth Acad. Conf. Chin. Assoc. Automat.*, 2016.

[31] B.-K. Ang, D. Dahlmeier, Z. Lin, J. Huang, M.-L. Seeto, and H. Shi, ''Indoor next location prediction with Wi-Fi,'' presented at the 4th Int. Conf. Digit. Inf. Process. Commun., 2014.

[32] L. Qiang, W. Shu, W. Liang, and T. Tan, ''Predicting the next location: A recurrent model with spatial and temporal contexts,'' in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 194–200.

[33] Y. Di, Z. Chao, J. Huang, and J. Bi, "SERM: A recurrent model for next location prediction in semantic trajectories," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 2411–2414, doi: 10.1145/3132847.3133056.

[34] J. C. Ying, W. C. Lee, and V. S. Tseng, "Mining geographic-temporal-semantic patterns in trajectories for location prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, pp. 1–33, 2014, doi: 10.1145/2542182.2542184.

[35] M. Sepahkar and M. R. Khayyambashi, "A novel collaborative approach for location prediction in mobile networks," *Wireless Netw.*, vol. 24, no. 1, pp. 283–294, Jan. 2018, doi: 10.1007/s11276-016-1304-1.

[36] D. Zhang, D. Zhang, H. Xiong, L. T. Yang, and V. Gauthier, "NextCell: Predicting location using social interplay from cell phone traces," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 452–463, Feb. 2015, doi: 10.1109/TC.2013.223.

[37] L. Wen, S. X. Xia, L. Feng, and Z. Lei, "Improving location prediction by exploring spatial-temporal-social ties," *Math. Problems Eng.*, vol. 2014, no. 4654, pp. 1–7, 2014, doi: 10.1155/2014/151803.

[38] W. Peixiao, W. Sheng, Z. Hengcai, and L. Feng, "Indoor location prediction method for shopping malls based on location sequence similarity," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 11, p. 517, 2019, doi: 10.3390/ijgi8110517.

[39] S. Lee, J. Lim, J. Park, and K. Kim, "Next place prediction based on spatiotemporal pattern mining of mobile device logs," *Sensors*, vol. 16, no. 2, p. 145, 2016, doi: 10.3390/s16020145.

[40] L. Vu, Q. Do, and K. Nahrstedt, "Jyotish: Constructive approach for context predictions of people movement from joint WiFi/Bluetooth trace," *Pervasive Mobile Comput.*, vol. 7, no. 6, pp. 690–704, 2011, doi: 10.1016/j.pmcj.2011.07.004.

[41] L. Vu, P. Nguyen, K. Nahrstedt, and B. Richerzhagen, "Characterizing and modeling people movement from mobile phone sensing traces," *Pervasive Mobile Comput.*, vol. 17, pp. 220–235, Feb. 2015, doi: 10.1016/j.pmcj.2014.12.001.

[42] T. M. T. Do, O. Dousse, M. Miettinen, and D. Gatica-Perez, "A probabilistic kernel method for human mobility prediction with smartphones," *Pervasive Mobile Comput.*, vol. 20, pp. 13–28, Jul. 2015, doi: 10.1016/j.pmcj.2014.09.001.

[43] F. Wu, K. Fu, Y. Wang, Z. Xiao, and X. Fu, "A spatial-temporal-semantic neural network algorithm for location prediction on moving objects," *Algorithms*, vol. 10, no. 2, p. 37, 2017, doi: 10.3390/a10020037.

[44] Y. Zhou, H. Sun, J. Huang, Z. Sun, H. Xiong, S. Qiao, Z. Guan, and X. Jia, "Efficient destination prediction based on route choices with transition matrix optimization," *IEEE Trans. Knowl. Data Eng.*, to be published, doi: 10.1109/TKDE.2018.2883938.

[45] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie, "Mining individual life pattern based on location history," in *Proc. IEEE Int. Conf. Mobile Data Manage.*, May 2009, pp. 1–10, doi: 10.1109/MDM.2009.11.

[46] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," *Data Knowl. Eng.*, vol. 65, no. 1, pp. 126–146, 2008, doi: 10.1016/j.datak.2007.10.008.

[47] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, 2007, doi: 10.1016/j.datak.2006.01.013.

[48] D. Xia, B. Wang, H. Li, Y. Li, and Z. Zhang, "A distributed spatial-temporal weighted model on MapReduce for short-term traffic flow forecasting," *Neurocomputing*, vol. 179, pp. 246–263, Feb. 2016.

[49] X. G. Yu, Y. H. Liu, W. Da, and L. Y. Lei, "A hybrid Markov model based on EM algorithm," in *Proc. Int. Conf. Control*, 2006.

[50] Y. Yang, "An evaluation of statistical approaches to text categorization," *Inf. Retr.*, vol. 1, nos. 1–2, pp. 69–90, 1999.

**PEIXIAO WANG** is currently pursuing the M.S. degree with The Academy of Digital China, Fuzhou University. His interests focus on indoor location services and spatial-temporal data mining.

**HONGEN WANG** is currently pursuing the M.S. degree with the College of Geomatics, Shandong University of Science and Technology. His interests focus on indoor location services and spatial-temporal data mining.

**HENGCAI ZHANG** received the Ph.D. degree from the Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences. He is currently an Assistant Professor with the Institute of Geographical Sciences and Natural Resources Research (IGSNRR), Chinese Academy of Sciences (CAS). His interests focus on moving objects database and spatial-temporal data mining. During the past years, he has published over 26 refereed journal articles and conference papers. He is a member of the Theory and Methodology Committee of the Chinese Association of Geographic Information System, and the Chinese Branch of ACM SIGSPATIAL.

**FENG LU** received the B.S. degree from Wuhan University and the Ph.D. degree from the Institute of Remote Sensing Applications, Chinese Academy of Sciences. He is currently a Professor with the Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences. He is also the Deputy Director of the State Key Laboratory of Resources and Environmental Information System and a Guest Ph.D. Advisor for Fuzhou University. During the past years, he has published over 120 refereed journal articles and conference papers. His research interests cover spatial data modeling, spatial DBMS, trajectory data mining, GIS for transportation, and urban GIS application. He is a member of the Information Technology Committee of Chinese Transportation Association, the Theory and Methodology Committee of the Chinese Association of GIS, and the New Technology Applications Committee of Chinese City Planning Association.

**SHENG WU** received the Ph.D. degree from the Institute of Surveying and Mapping, Information Engineering University, Zhengzhou. He is currently a Professor with Fuzhou University. He is also the Deputy Director of the Academy of Digital China. During the past years, he has published over 60 refereed journal articles and conference papers. His interests focus on spatial-temporal data mining and visualization, and digital government. He is a member of the Theory and Methodology Committee of the Chinese Association of Geographic Information System and the Cartography and GIS Committee of China Society of Surveying and mapping.